

⑯ Aktenzeichen: 100 50 172.9
⑯ Anmeldetag: 11. 10. 2000
⑯ Offenlegungstag: 26. 4. 2001

⑯ Unionspriorität:
09/419841 15. 10. 1999 US
⑯ Anmelder:
International Business Machines Corp., Armonk,
N.Y., US
⑯ Vertreter:
Duscher, R., Dipl.-Phys. Dr.rer.nat., Pat.-Ass., 71034
Böblingen

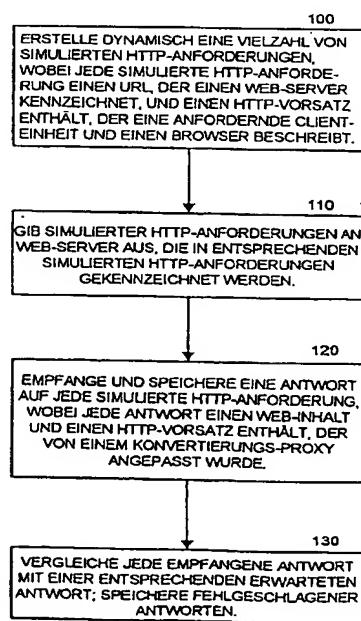
⑯ Erfinder:
Camut, Samuel A., Apex, N.C., US; Heninger, Ivan
M., Selma, N.C., US; Vought, Eric, Raleigh, N.C., US

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

Prüfungsantrag gem. § 44 PatG ist gestellt

⑯ Systeme, Verfahren und Computerprogrammprodukte zur Überprüfung eines für die Anzeige in pervasive Computereinheiten angepassten Web-Inhaltes

⑯ Es werden Systeme, Verfahren und Computerprogrammprodukte bereitgestellt, um zu prüfen, ob ein Web-Inhalt von einem Konvertierungs-Proxy für die Anzeige in verschiedenen anfordernden pervasive Datenverarbeitungseinheiten ordnungsgemäß angepasst wurde. Unter Verwendung von Informationen aus einer oder mehreren Datendateien werden simulierte HyperText Transfer Protocol- (HTTP-) Anforderungen erstellt. Jede simulierte Anforderung enthält einen Uniform Resource Locator (URL), der eine Position des Web-Inhaltes kennzeichnet. Außerdem enthält jede simulierte Anforderung einen HTTP-Vorsatz, der Informationen über eine entsprechende pervasive Datenverarbeitungseinheit enthält. Simulierte HTTP-Anforderungen werden asynchron zu entsprechenden Web-Servern ausgegeben, die in den jeweiligen HTTP-Anforderungen gekennzeichnet werden. Eine HTTP-Antwort auf jede entsprechende simulierte HTTP-Anforderung wird empfangen und enthält einen Web-Inhalt, der für die Anzeige in einer entsprechenden pervasive Datenverarbeitungseinheit angepasst wurde, die der jeweiligen simulierten HTTP-Anforderung zugeordnet ist. Jede HTTP-Antwort wird sodann mit einer erwarteten HTTP-Antwort verglichen. Eine HTTP-Antwort, deren Vergleich mit einer erwarteten HTTP-Antwort ungünstig ausfällt, kann zur späteren Analyse gespeichert werden.



Beschreibung

Gebiet der Erfindung

Die vorliegende Erfindung betrifft im Allgemeinen die Anzeige des Inhaltes einer Web-Seite und insbesondere die Anpassung des Web-Inhaltes für die Anzeige in Client-Einheiten.

Grundlagen der Erfindung

Das Internet ist ein weltweites dezentrales Netz von Computern mit der Fähigkeit, miteinander zu kommunizieren. Das Internet hat umfassende Anerkennung als leistungsfähiges Medium zur Kommunikation und Interaktion über mehrere Netze erlangt. Das World-Wide Web (Web) ist Anfang der 90er Jahre entstanden und besteht aus mit dem Internet verbundenen Server-Hostcomputern (Web-Server), die Hypertextdokumente (als Web-Seiten bezeichnet) darin gespeichert haben. Client-Programme (z. B. Web-Browser), die das Hypertext Transfer Protocol (HTTP) verwenden, können über eine Transmission Control Protocol/Internet Protocol- (TCP/IP-) Verbindung zwischen einer Client-Hosteinheit (client-hosting device) und einer Server-Hosteinheit (server-hosting device) auf Web-Seiten zugreifen. Obwohl HTTP und Web-Seiten die vorherrschenden Formen für das Web sind, betrifft das Web selbst einen weiten Bereich von Protokollen, darunter Secure Hypertext Transfer Protocol (HTTPS), File Transfer Protocol (FTP) und Gopher, und Formaten von Web-Inhalten, darunter unverschlüsselter Text, HyperText Markup Language (HTML), Extensible Markup Language (XML), sowie Bildformate, beispielsweise Graphics Interchange Format (GIF) und Joint Photographic Experts Group (JPEG).

Eine Web-Seite ist herkömmlicherweise eine Sammlung von Web-Seiten und anderen, ein bestimmtes Thema betreffenden Dateien, die eine als "Home"-Page bezeichnete Anfangsdatei enthält. Eine große Web-Seite kann sich auf einer Anzahl geographisch verstreuter Web-Server befinden. Die Web-Seite von International Business Machines Corporation (www.ibm.com) besteht beispielsweise aus Tausenden von Web-Seiten und Dateien, die weltweit über mehrere Web-Server verteilt sind.

Ein Intranet ist ein privates Computernetz, das herkömmlicherweise in einem Unternehmen enthalten ist und das herkömmlicherweise einen oder mehrere Server enthält, die mit mehreren Benutzercomputern kommunizieren. Ein Intranet kann aus miteinander verbundenen lokalen Netzen (local area networks) bestehen und kann außerdem Mietleitungen in einem landesweiten Netz (wide-area network) verwenden. Ein Intranet kann Verbindungen zum äußeren Internet enthalten. Intranets verwenden herkömmlicherweise mehrere Internet-Protokolle und erscheinen im Allgemeinen wie private Versionen des Internets. Ein Intranetbenutzer greift normalerweise über einen lokal auf seinem/ihrem Computer laufenden Browser auf einen Intranet-Server zu.

Ein Web- (oder Intranet-) Server ist ein (normalerweise auf einem Computer ablaufendes) Computerprogramm, das angeforderte Web-Seiten und Dateien bedient. Ein Web-Client ist ein anforderndes Programm, das einem Benutzer zugeordnet ist. Ein Browser ist ein exemplarischer Web-Client zur Verwendung bei der Anforderung von Web-Seiten und Dateien von Web-Servern. Ein Web-Server wartet auf einen Web-Client, beispielsweise einen Browser, um eine Verbindung zu eröffnen und eine spezifische Web-Seite (oder Datei) anzufordern. Der Web-Server sendet sodann eine Kopie des angeforderten Elementes, schließt die Verbindung und wartet auf die nächste Verbindung.

Wenn ein Browser mit einem Web-Server in Wechselwirkung steht, verwenden die beiden Programme HTTP. HTTP ermöglicht es einem Browser, ein spezifisches Element anzufordern, das der Web-Server sodann zurücksendet und der

5 Browser wiedergibt. Zur Sicherstellung einer unzweideutigen Kooperation von Browsern und Web-Servern definiert HTTP das genaue Format von Anforderungen (HTTP-Anforderungen), die von einem Browser zu einem Web-Server gesendet werden, sowie das Format von Antworten (HTTP- Antworten), die der Web-Server zum Browser zurücksendet.

Exemplarische Browser für Internet und Intranet beinhalten Netscape Navigator® (America Online, Inc., 22000 AOL Way, Dulles, VA) und Internet Explorer® (Microsoft Corporation, Redmond, WA). Browser stellen normalerweise eine grafische Benutzerschnittstelle zum Abrufen und Betrachten von Web-Seiten, Anwendungen und anderen Ressourcen bereit, die sich auf Internet/Intranet-Servern (hierin im Folgenden kollektiv als "Web-Server" bezeichnet) befinden.

10 20 Wie Fachleuten bekannt ist, wird eine Web-Seite herkömmlicherweise über eine Standardseitenbeschreibungssprache (standard page description language), zum Beispiel HTML, formatiert, die normalerweise Text enthält und sich auf Grafiken, Ton, Animation und Videodaten beziehen

25 30 35 kann. HTML schafft die Voraussetzungen für eine grundlegende Dokumentformatierung und ermöglicht einem Web-Inhalt-Provider die Angabe spezifischer Anker oder Hypertextverbindungen (normalerweise als hervorgehobener Text gezeigt) zu anderen Web-Servern und Dateien. Wenn ein Benutzer eine bestimmte Hypertextverbindung auswählt, liest und interpretiert ein Browser eine der Verbindung zugeordnete Adresse, die als Uniform Resource Locator (URL) bezeichnet wird, verbindet den Browser mit einem Web-Server bei dieser Adresse und erstellt eine Anforderung (z. B. eine HTTP-Anforderung) für die in der Verbindung gekennzeichnete Datei. Der Web-Server sendet die angeforderte Datei sodann zum Web-Client, den der Browser interpretiert und dem Benutzer anzeigt.

Viele neue elektronische Einheiten, beispielsweise Personal Digital Assistants (PDAs), Taschencomputer und Web-TVs, erhalten als Client-Einheiten Zugriff auf das Internet und/oder auf Intranets. Elektronische Einheiten, darunter jedoch nicht ausschließlich – PDAs, Zelltelefone (cellular telephones) und Datenverarbeitungseinheiten, die in Geräten und Kraftfahrzeugen verwendet werden, werden oftmals insgesamt als "pervasive" Datenverarbeitungseinheiten bezeichnet. Viele solcher pervasive Datenverarbeitungseinheiten verwenden die Plattformen von Microsoft® Windows CE und 3Com Palm Computing®.

40 45 50 Leider sind die Fähigkeiten von pervasive Datenverarbeitungseinheiten zum Empfangen, Verarbeiten, Speichern und Anzeigen eines Internet-Inhaltes unterschiedlich. Die Anzeigen von pervasive Datenverarbeitungseinheiten sind im Vergleich zu Tischcomputeranzeigen (desktop computer displays) klein. Infolgedessen können Inhaltsanteile einer Web-Seite, beispielsweise Bilder und konvertierte HTML, die andernfalls auf einer Tischcomputeranzeige angezeigt werden können, auf der Anzeige einer pervasive Computer-Einheit nicht angezeigt werden, wenn nicht einige Änderungen an den Bildern und/oder dem Text (d. h. dem Inhalt) vorgenommen werden. Beispielsweise kann eine Tischcomputeranzeige mit einer Matrix von 1024 Bildelementen mal 768 Bildelementen ein großes (z. B. 2 MB) Farbbild mit 24 Bit pro Bildelement anzeigen. Eine pervasive Computereinheit, die eine Anzeige mit einer Matrix von 120 Bildelementen mal 120 Bildelementen aufweist und nur etwa 3 Bit pro Bildelement anzeigen kann, ignoriert möglicherweise viele der Bilddaten. Infolgedessen kann das Bild über die perva-

sive Computereinheit nicht ordnungsgemäß angezeigt werden, falls überhaupt, es sei denn, die Größe des Bildes wird an die Fähigkeiten der Datenverarbeitungseinheit angepasst. Außerdem können einige pervasive Datenverarbeitungseinheiten bestimmte Bilddateitypen, beispielsweise JPEG oder GIF, möglicherweise nicht anzeigen.

Die Schriftarten und Größen von Text im Web-Inhalt müssen möglicherweise außerdem geändert werden, um deren Anzeige in der Anzeige einer pervasive Datenverarbeitungseinheit zu ermöglichen. Außerdem können allgemeine HTML-Merkmale, zum Beispiel Rahmen und Tabellen, von pervasive Datenverarbeitungseinheiten möglicherweise nicht angezeigt werden. Daten in Rahmen und Tabellen müssen zum korrekten Anzeigen möglicherweise entfernt und/oder in andere Konfigurationen neu formatiert werden. Außerdem erfordern Leistungsbegrenzungen von pervasive Datenverarbeitungseinheiten, beispielsweise die Speichergröße und die Verbindungsbandbreite, möglicherweise ebenfalls Änderungen am Web-Seiteninhalt, um diesen über eine pervasive Datenverarbeitungseinheit korrekt anzuzeigen.

Es ist bekannt, einen Web-Inhalt, der über eine pervasive Datenverarbeitungseinheit nicht korrekt angezeigt werden kann, in ein anzeigbares Format "anzupassen". Große Farbbilder mit hoher Auflösung können beispielsweise in kleine Schwarzweißbilder umgesetzt werden, die in kleinen Anzeigen mit niedriger Auflösung angezeigt werden können. Die Anpassung von Video, Bildern, Ton und Text zur Anzeige in einer Client-Einheit wird normalerweise als Konvertierung ("transcoding") bezeichnet. Eine Web-Inhalt-Konvertierung wird in den schwebenden US-Patentanmeldungen Nr. 09/239 935 und 09/240 137 ausführlich beschrieben, die International Business Machines Corporation zugewiesen und in ihrer Gesamtheit hierin durch Bezugnahme aufgenommen werden. Die Web-Inhalt-Konvertierung wird normalerweise von einem Konvertierungs-Proxy (transcoding proxy) ausgeführt, der einem Web-Server zugeordnet ist. Produkte zur Konvertierung eines Web-Inhaltes zur Anzeige durch eine anfordernde Client-Einheit sind bekannt. Ein Beispiel für ein Konvertierungsprodukt ist das von der Apache Server Foundation unter <http://java.apache.org> erhältliche "Cocoon". Cocoon wird unter <http://java.apache.org/cocoon/index.html> beschrieben, das durch Bezugnahme hierin aufgenommen wird.

Zum Ausführen einer Web-Inhalt-Konvertierung muss ein Web-Server und/oder ein Konvertierungs-Proxy normalerweise Kenntnisse von einer Client-Einheit haben, die eine HTTP-Anforderung erstellt. Wie Fachleuten bekannt ist, begleitet ein HTTP-Vorsatz (HTTP header) HTTP-Anforderungen zu einem Web-Server. Ein HTTP-Vorsatz stellt normalerweise Informationen über eine anfordernde Client-Einheit und deren Browser bereit. Beispiele für in einem HTTP-Vorsatz bereitgestellte Informationen können Folgendes enthalten: die Anzeigegröße einer Client-Einheit, ob die Anzeige einer Client-Einheit eine Farbanzeige oder eine Schwarzweißanzeige ist, eine Kennung des Browsers der Client-Einheit und eine Kennung des Betriebssystems der Client-Einheit.

Es werden laufend neue Datenverarbeitungseinheiten für das breite Publikum mit verschiedenen Browsern und Konfigurationen entwickelt und auf den Markt gebracht. Unter Berücksichtigung der Vielfalt neuer Datenverarbeitungseinheiten für die Allgemeinheit kann es für die Herausgeber von Web-Inhalten schwierig sein, den Inhalt so anzupassen, dass er in vielen verschiedenen Einheiten angemessen angezeigt werden kann. Infolgedessen besteht ein Bedarf an einer schnellen und problemlosen Überprüfung, dass die Anpassung eines Web-Inhaltes für jeden Typ von Client-Ein-

heit, die eine HTTP-Anforderung sendet, korrekt ausgeführt wird. Gegenwärtig erfolgt die Überprüfung, ob die Anpassung des Web-Inhaltes korrekt ausgeführt wird, durch das Senden von HTTP-Anforderungen mit jeder tatsächlichen

5 Client-Einheit. Leider kann die Fähigkeit zum Prüfen der Web-Inhaltanpassung mit vielen verschiedenen Client-Einheiten kostspielig und zeitaufwendig sein, und es ist möglicherweise technisch nicht durchführbar, Antworten analytisch auf Gültigkeit zu prüfen.

10

Zusammenfassung der Erfindung

Angesichts der obigen Erläuterung ist es eine Aufgabe der vorliegenden Erfindung, Systeme, Verfahren und Computerprogrammprodukte zum Simulieren verschiedener, auf unterschiedlichen Softwareplattformen betriebener Client-Einheiten bereitzustellen, um sicherzustellen, dass die Anpassung eines Web-Inhaltes korrekt ausgeführt wird.

Eine andere Aufgabe der vorliegenden Erfindung ist das 20 Erleichtern der Anzeige von Web-Seiten über pervasive Datenverarbeitungseinheiten, die im Vergleich zu Tischcomputern kleinere Anzeigen und verschiedene Leistungsbegrenzungen aufweisen können.

Diese und andere Aufgaben der vorliegenden Erfindung 25 werden von Systemen, Verfahren und Computerprogrammprodukten bereitgestellt, um zu testen (d. h. zu überprüfen), ob der Web-Inhalt von einem Konvertierungs-Proxy für die Anzeige in verschiedenen anfordernden pervasive Datenverarbeitungseinheiten ordnungsgemäß angepasst wurde.

30 Unter Verwendung von Informationen von einer oder mehreren Datendateien werden simulierte HyperText Transfer Protocol-(HTTP-) Anforderungen erstellt. Jede simulierte Anforderung enthält vorzugsweise einen Uniform Resource Locator (URL), der eine Position des Web-Inhaltes kennzeichnet. Außerdem enthält jede simulierte Anforderung vorzugsweise einen HTTP-Vorsatz, der Informationen über eine entsprechende pervasive Datenverarbeitungseinheit enthält. Beispiele für in einem HTTP-Vorsatz enthaltene Informationen über pervasive Datenverarbeitungseinheiten 35 können die folgenden Informationen enthalten, sind jedoch nicht auf diese begrenzt: eine Kennung des Browsers einer Einheit, eine Kennung des Betriebssystems einer Einheit, Eigenschaften der Anzeige einer Einheit und Informationen über den Web-Inhalt, für dessen Anzeige eine Einheit konfiguriert ist.

Simulierte HTTP-Anforderungen werden zu entsprechenden Web-Servern ausgegeben, die in den jeweiligen HTTP-Anforderungen gekennzeichnet werden. Auf jede entsprechende simulierte HTTP-Anforderung wird eine HTTP-50 Antwort empfangen, die einen Web-Inhalt enthält, der zur Anzeige in einer entsprechenden pervasive Datenverarbeitungseinheit angepasst wurde, die der jeweiligen simulierten HTTP-Anforderung zugeordnet ist. Jede HTTP-Antwort wird sodann mit einer erwarteten HTTP-Antwort verglichen. Eine HTTP-Antwort, deren Vergleich mit einer erwarteten HTTP-Antwort keine Übereinstimmung ergibt, kann 55 zur späteren Analyse gespeichert werden.

Die vorliegende Erfindung kann die Feststellung erleichtern, ob ein Web-Inhalt für die Anzeige in fast jedem Typ

60 von Client-Einheit mit fast jedem Konfigurationstyp ordnungsgemäß angepasst ist, ohne dass die eigentliche Einheit im Test verwendet werden muss. Infolgedessen kann die Verwendung der vorliegenden Erfindung von Web-Inhalt-Providern zu beträchtlichen Zeiteinsparungen, Kosteneinsparungen und Programmprodukten von höherer Qualität 65 führen.

Kurze Beschreibung der Zeichnungen

Fig. 1 veranschaulicht schematisch die Pfade einer HTTP-Anforderung und einer entsprechenden HTTP-Antwort in einer Client-Server-Umgebung, die einen Konvertierungs-Proxy enthält, der einen Web-Inhalt für die Anzeige in anfordernden Client-Einheiten anpasst.

Fig. 2 ist ein Flussdiagramm, das Arbeitsgänge zum Überprüfen eines konvertierten Web-Inhaltes darstellt, der gemäß der vorliegenden Erfundung auf eine Client-Anforderung hin bereitgestellt wird.

Fig. 3 ist ein ausführliches Flussdiagramm, das Arbeitsgänge zum Erstellen und Ausgeben einer Vielzahl von HTTP-Anforderungen und zum Überprüfen eines konvertierten Web-Inhaltes darstellt, der gemäß der vorliegenden Erfundung auf eine Client-Anforderung hin bereitgestellt wird.

Fig. 4 ist eine schematische Darstellung eines Datenverarbeitungssystems zum Ausführen von Arbeitsgängen gemäß der vorliegenden Erfundung.

Fig. 5 ist eine schematische Darstellung eines Datenverarbeitungssystems, das mit einer Vielzahl von Web-Servern und Konvertierungs-Proxies kommuniziert, wobei das Datenverarbeitungssystem so konfiguriert ist, dass es gemäß der vorliegenden Erfundung eine Vielzahl von HTTP-Anforderungen erstellt und sendet und eine entsprechende Vielzahl von HTTP-Antworten empfängt.

Ausführliche Beschreibung der Erfundung

Im Folgenden wird die vorliegende Erfundung nun ausführlicher mit Bezugnahme auf die begleitenden Zeichnungen beschrieben, in denen bevorzugte Ausführungsformen der Erfundung gezeigt werden. Diese Erfundung kann jedoch in vielen verschiedenen Formen ausgeführt werden und sollte nicht als auf die hierin dargelegten Ausführungsformen begrenzt gedeutet werden; stattdessen werden diese Ausführungsformen bereitgestellt, damit diese Beschreibung sorgfältig und vollständig ist und Fachleuten den Anwendungsbereich der Erfundung vollständig zeigt. Gleiche Ziffern beziehen sich durchweg auf gleiche Elemente.

Wie Fachleute verstehen werden, kann die vorliegende Erfundung als Verfahren, Datenverarbeitungssystem oder Computerprogrammprodukt ausgeführt werden. Dementsprechend kann die vorliegende Erfundung die Form einer kompletten Hardware-Ausführungsform, einer kompletten Software-Ausführungsform oder einer Ausführungsform annehmen, die Software- und Hardwareaspekte kombiniert. Außerdem kann die vorliegende Erfundung die Form eines Computerprogrammproduktes auf einem in einem Computer verwendbaren Speichermedium annehmen, das ein in einem Computer verwendbares, im Medium integriertes Programmcodemittel aufweist. Jedes geeignete computerlesbare Medium kann verwendet werden, darunter Festplatten, CD-ROMs, optische Speichereinheiten oder magnetische Speichereinheiten.

Der Computerprogrammcode zum Ausführen von Arbeitsgängen der vorliegenden Erfundung wird vorzugsweise in einer objektorientierten Programmiersprache, beispielsweise JAVA®, Smalltalk oder C++, geschrieben. Der Computerprogrammcode zum Ausführen von Arbeitsgängen der vorliegenden Erfundung kann jedoch auch in herkömmlichen prozeduralen Programmiersprachen (procedural programming languages), beispielsweise der Programmiersprache "C", oder in einer funktionellen Programmiersprache (oder Programmiersprache der vierten Generation), beispielsweise Lisp, SML oder Forth, geschrieben werden.

Die vorliegende Erfundung wird unten mit Bezugnahme

auf Blockschaltbilder und/oder Flussdiagrammdarstellungen von Verfahren, Vorrichtungen (Systemen) und Computerprogrammprodukten gemäß einer Ausführungsform der Erfundung beschrieben. Es ist klar, dass jeder Block der 5 Blockschaltbilder und/oder Flussdiagrammdarstellungen durch Computerprogrammbefehle realisiert werden kann. Diese Computerprogrammbefehle können einem Prozessor eines Universalcomputers, eines Spezialcomputers oder einer anderen programmierbaren Datenverarbeitungsvorrichtung 10 zur Erzeugung einer Maschine zugeführt werden, so dass die Befehle, die über den Prozessor des Computers oder der anderen programmierbaren Datenverarbeitungsvorrichtung ablaufen, Mittel zum Realisieren der im Blockschaltbild und/oder Flussdiagrammblöcken angegebenen Funktionen erzeugen.

Diese Computerprogrammbefehle können außerdem in einem computerlesbaren Speicher gespeichert werden, der einen Computer oder eine andere programmierbare Datenverarbeitungsvorrichtung auf eine bestimmte Weise steuern 15 kann, so dass die im computerlesbaren Speicher gespeicherten Befehle ein Fertigungsprodukt erzeugen, darunter Befehlsmittel, die im Blockschaltbild und/oder Flussdiagrammblöcken angegebene Funktion realisieren.

20 Die Computerprogrammbefehle können außerdem in einen Computer oder eine andere programmierbare Datenverarbeitungsvorrichtung geladen werden, um zu bewirken, dass eine Reihe von Verarbeitungsschritten auf dem Computer oder der anderen programmierbaren Vorrichtung ausgeführt werden, um einen in einem Computer realisierten Prozess 25 zu erzeugen, so dass die Befehle, die auf dem Computer oder der anderen programmierbaren Vorrichtung ablaufen, Schritte zur Realisierung der in den Blockschaltbildern und/oder im Flussdiagrammblöcken angegebenen Funktionen bereitstellen.

Mit Bezugnahme auf **Fig. 1** werden nun die Pfade einer HTTP-Anforderung und einer entsprechenden HTTP-Antwort in einer Client-Server-Beziehung schematisch dargestellt, wobei ein Konvertierungs-Proxy zum Anpassen des 30 Web-Inhaltes enthalten ist. Eine Client-Einheit 10 sendet eine HTTP-Anforderung 15 nach einem Web-Inhalt von einem HTTP- (d. h. Web-) Server 20. Das Protokoll für HTTP-Anforderungen und -Antworten wird in der Veröffentlichung "Hypertext Transfer Protocol – HTTP/1.1", Network Working Group Request for Comments 2068, Januar 1997, die in ihrer Gesamtheit durch Bezugnahme hierin aufgenommen ist, ausführlich beschrieben.

In der dargestellten Client-Server-Beziehung durchläuft die HTTP-Anforderung 15 einen Konvertierungs-Proxy 30 und wird vom Konvertierungs-Proxy 30 geändert, wie Fachleute verstehen würden. Die geänderte HTTP-Anforderung 15' wird vom HTTP-Server 20 verarbeitet, und eine den angeforderten Web-Inhalt enthaltende HTTP-Antwort 25 wird über den Konvertierungs-Proxy 30 zur Client-Einheit 10 gesendet. Der Konvertierungs-Proxy 30 passt den Web-Inhalt in der HTTP-Antwort 25 an, wobei eine "geänderte" HTTP-Antwort 25' erzeugt wird. Der angepasste Web-Inhalt in der geänderten HTTP-Antwort 25' kann in der Client-Einheit 10 wiedergegeben werden. Es ist klar, dass ein Konvertierungs-Proxy 35 30 als im HTTP-Server 20 befindlicher Softwarecode, als außerhalb des HTTP-Servers, beispielsweise innerhalb einer Firewall, befindlicher Softwarecode oder einer Kombination aus beidem realisiert werden kann.

Die vorliegende Erfundung stellt Verfahren, Systeme und Computerprogrammprodukte zur Überprüfung eines konvertierten Web-Inhaltes bereit, der auf HTTP-Anforderungen 40 hin bereitgestellt wurde, ohne dass die eigentlichen Client-Einheiten HTTP-Anforderungen senden müssen. Ge-

mäß der vorliegenden Erfindung können mehrere Client-Anforderungen von einem einzigen Computer simuliert werden, der beispielsweise über das Internet, ein Intranet oder ein anderes Datenübertragungsnetz mit einem oder mehreren Web-Servern kommuniziert.

Mit Bezugnahme auf **Fig. 2** werden Arbeitsgänge zum Überprüfen eines konvertierten Web-Inhaltes gemäß der vorliegenden Erfindung dargestellt. Unter Verwendung verschiedener Datendateien werden von einer Datenverarbeitungseinheit eine Vielzahl simulierter HTTP-Anforderungen erstellt (Block 100). Jede simulierte HTTP-Anforderung enthält einen Uniform Resource Locator (URL), der die Position eines spezifischen Web-Inhaltes kennzeichnet, der in einer anfordernden Client-Einheit angezeigt werden soll. Jede HTTP-Anforderung enthält außerdem einen HTTP-Vorsatz, der Informationen über eine anfordernde Client-Einheit enthält und der die Typen von Web-Inhalt angibt, auf deren Verarbeitung die Client-Einheit vorbereitet ist. Wie Fachleuten bekannt ist, übertragen HTTP-Vorsätze Informationen, die Web-Server und Browser zur Definition des zwischen ihnen ausgetauschten Datentyps verwenden.

Gleichzeitige HTTP-Anforderungen werden vorzugsweise asynchron an Web-Server ausgegeben, die in den jeweiligen HTTP-Anforderungen gekennzeichnet werden (Block 110, **Fig. 2**).

Eine Datenverarbeitungseinheit zur Realisierung der vorliegenden Erfindung setzt vorzugsweise für jede der betreffenden HTTP-Anforderungen das HTTP-"GET"-Verfahren (HTTP "GET" Method) ein. Das HTTP-"GET"-Verfahren, das Fachleuten gut bekannt ist, wird in "Hypertext Transfer Protocol – HTTP/1.1", Network Working Group Request for Comments 2068, Januar 1997, ausführlich beschrieben.

Eine HTTP-Antwort auf jede HTTP-Anforderung wird vorzugsweise gespeichert (Block 120, **Fig. 2**). Jede HTTP-Antwort enthält einen HTTP-Vorsatz und einen Web-Inhalt, der für die Anzeige in einer Client-Einheit angepasst wurde, die als die Einheit identifiziert wurde, die die entsprechende HTTP-Anforderung sendete. Jede empfangene HTTP-Antwort wird mit einer erwarteten HTTP-Antwort verglichen (Block 130, **Fig. 2**). HTTP-Antworten, die nicht mit ihren jeweiligen erwarteten Antworten übereinstimmen, werden vorzugsweise zur späteren Analyse gespeichert.

Wie in **Fig. 3** dargestellt wird, können mehrere simulierte HTTP-Anforderungen gleichzeitig und asynchron bis zu "n" Wiederholungen erstellt und ausgegeben werden, wobei n praktisch ein beliebiger Wert sein kann. Die gleichzeitige Erstellung und Ausgabe mehrerer simulierter HTTP-Anforderungen werden durch die Arbeitsgänge 102-1 bis 130-1, die Arbeitsgänge 102-2 bis 130-2 und die Arbeitsgänge 102-n bis 130-n schematisch dargestellt. Eine ausführliche Beschreibung der Arbeitsgänge 102-1 bis 130-1 wird unten bereitgestellt, und es ist klar, dass diese Arbeitsgänge für alle der n Wiederholungen gleich sind.

Für jede HTTP-Anforderung wählt eine die vorliegende Erfindung realisierende Datenverarbeitungseinheit vorzugsweise einen URL aus, der den aus einer Datendatei abzurufenden Web-Inhalt definiert (Block 102-1). Ähnlich werden HTTP-Vorsatzfelder vorzugsweise aus einer Datendatei ausgewählt (Block 104-1). Ein ausgewählter URL und ausgewählte HTTP-Vorsatzfelder werden sodann zum Erstellen jeder betreffenden HTTP-Anforderung kombiniert (Block 106-1). Jede erstellte HTTP-Anforderung wird sodann zum Web-Server ausgegeben, der in der erstellten HTTP-Anforderung gekennzeichnet wird (Block 110-1).

Nachdem eine Antwort auf eine HTTP-Anforderung empfangen und gespeichert wurde (Block 120-1), wird eine Entscheidung getroffen, ob eine weitere HTTP-Anforderung erstellt werden soll (Block 122-1). Sobald eine Ent-

scheidung getroffen wurde, keine weitere HTTP-Anforderung zu erstellen, wird jede zurückgesendete HTTP-Antwort mit einer erwarteten Antwort verglichen (Block 130-1). Vorzugsweise wird jede fehlgeschlagene Antwort zur späteren Analyse gespeichert, um festzustellen, warum der Web-Inhalt nicht ordnungsgemäß konvertiert wurde.

Die Entscheidung zum Erstellen einer weiteren HTTP-Anforderung kann von dem in einer HTTP-Antwort zurückgesendeten Web-Inhalt abhängen. Falls der Web-Inhalt in einer HTTP-Antwort beispielsweise Bilder enthält, kann die vorliegende Erfindung entsprechende HTTP-Anforderungen für jedes Bild erstellen. Dementsprechend können die Verfahren der vorliegenden Erfindung rekursiv sein, und die sich wiederholenden Schritte von **Fig. 3** können fortgesetzt werden, bis alle Typen von Dateien in einer Web-Seite angefordert worden sind. Infolgedessen kann die Anpassung des Inhaltes einer ganzen Web-Site über ein Datenverarbeitungssystem überprüft werden, das die vorliegende Erfindung realisiert.

Beispielhafte Informationen über eine anfordernde Client-Einheit, die in HTTP-Vorsatzfeldern bereitgestellt wird, können Folgendes enthalten, sind jedoch nicht darauf begrenzt: eine Kennung eines auf der Client-Einheit (z. B. Benutzer-Agent) ablaufenden Browsers, eine Kennung des Betriebssystems der Client-Einheit, eine Kennung von Eigenschaften der Anzeige der Client-Einheit und eine Kennung des Web-Inhaltes, für dessen Anzeige die Client-Einheit konfiguriert ist. Beispielhafte Anzeigeeigenschaften einer Client-Einheit beinhalten – jedoch nicht ausschließlich – die Anzeigegröße und ob die Anzeige eine Farbanzeige oder eine Schwarzweißanzeige ist.

Mit Bezugnahme auf **Fig. 4** wird ein Datenverarbeitungssystem 600 zum Ausführen von Arbeitsgängen der vorliegenden Erfindung schematisch dargestellt. Wie in **Fig. 4** zu erkennen ist, kann ein Datenprozessor 601 ein Betriebssystem 602 aufweisen, das sich mit verschiedenen, auf dem Betriebssystem 602 ablaufenden Anwendungsprogrammen 603 zum Ausführen von Arbeitsgängen der vorliegenden Erfindung darin befindet. Insbesondere wird eine Anwendung 603a zur HTTP-Anforderungserstellung (HTTP Request Builder application) bereitgestellt, um die in den Blöcken 100 und 110 von **Fig. 2** und den Blöcken 100 bis 122-1 von **Fig. 3** dargelegten Arbeitsgänge zur HTTP-Anforderungserstellung und -ausgabe auszuführen. Eine Antwortvergleichs-Anwendung 603b (Response Comparator application) wird bereitgestellt, um Arbeitsgänge zum Vergleichen empfanger HTTP-Antworten mit erwarteten Antworten auszuführen, die im Block 130 von **Fig. 2** und im Block 130-1 von **Fig. 3** dargelegt werden.

Immer noch Bezug nehmend auf **Fig. 4** kommunizieren die Anwendungen 603a, 603b mit dem Datenspeicher 609, der ein externer oder ein interner Datenspeicher oder eine Kombination aus externem und internem Datenspeicher sein kann. Die Anwendung 603a zur HTTP-Anforderungserstellung ist so konfiguriert, dass sie HTTP-Vorsatzelemente und URLs aus dem Datenspeicher 609 liest und HTTP-Anforderungen erstellt und ausgibt, wie in den Blöcken 100 und 110 von **Fig. 2** und den Blöcken 100 bis 122-1 von **Fig. 3** dargestellt wird. Die Antwortvergleichs-Anwendung 603b ist so konfiguriert, dass empfangene HTTP-Antworten mit erwarteten Antworten verglichen werden, die im Block 130 von **Fig. 2** und im Block 130-1 von **Fig. 3** dargelegt werden. Außerdem kann der Datenspeicher 609 zur Speicherung erstellter HTTP-Anforderungen und zur Speicherung empfanger HTTP-Antworten und zur Realisierung der verschiedenen Verfahrensschritte der vorliegenden Erfindung als ein oder mehrere Stapelprozesse (batch processes) verwendet werden.

Wie Fachleute verstehen würden, zeigt der Prozessor 601 Informationen auf einer Anzeigeeinheit 604 an. Die Anzeigeeinheit 604 weist eine Vielzahl von Bildelementen (insgesamt als Bildschirm bezeichnet) auf, die das Erscheinen einer auf der Anzeigeeinheit 604 angezeigten grafischen Benutzeroberfläche (GUI) definieren können. Der Inhalt des Bildschirms und daher das Erscheinen einer GUI können von einem oder mehreren Anwendungsprogrammen 603a, 603b oder vom Betriebssystem 602, einzeln oder kombiniert, gesteuert oder geändert werden. Um eine Eingabe von einem Benutzer zu erhalten, kann das Betriebssystem 602, die Anwendungsprogramme 603a, 603b oder eine Kombination davon Benutzereingabeeinheiten 605 verwenden. Die Benutzereingabeeinheiten 605 können eine Zeigereinheit 606 und eine Tastatur 607 oder andere, Fachleuten bekannte Eingabeeinheiten beinhalten.

Mit Bezugnahme auf Fig. 5 wird nun ein Datenverarbeitungssystem 600 zum Ausführen von Arbeitsgängen der vorliegenden Erfindung dargestellt, das über ein Datenübertragungsnetz, beispielsweise das Internet und/oder ein Intranet 701, mit einer Vielzahl von Web-Servern 702 und Konvertierungs-Proxies 704 kommuniziert. In der dargestellten Ausführungsform können ein oder mehrere Web-Server 702 einen Konvertierungs-Proxy 704 verwenden, der zur Anpassung eines Web-Inhaltes gemäß der obigen Beschreibung konfiguriert ist. Das Datenverarbeitungssystem 600 ist gemäß der obigen Beschreibung so konfiguriert, dass es HTTP-Anforderungen von mehreren pervasive Datenverarbeitungseinheiten simuliert, ohne dass tatsächliche Datenverarbeitungseinheiten zum Erstellen von HTTP-Anforderungen benötigt werden. Das Datenverarbeitungssystem 600 erstellt mehrere HTTP-Anforderungen und gibt diese aus, wobei jede HTTP-Anforderung einen HTTP-Vorsatz enthält, der Informationen über eine entsprechende pervasive Client-Einheit/Browser-Konfiguration (client device/browser configuration) enthält. Diese HTTP-Anforderungen können sodann als asynchrone parallele Anforderungen an gekennzeichnete Web-Server gesendet werden.

Das Datenverarbeitungssystem 600 ist gemäß der obigen Beschreibung so konfiguriert, dass es über das Internet und/oder das Intranet 701 HTTP-Antworten auf die HTTP-Anforderungen empfängt und die HTTP-Antworten mit erwarteten HTTP-Antworten vergleicht. Vorzugsweise ist das Datenverarbeitungssystem 600 so konfiguriert, dass es HTTP-Antworten speichert und anschließend HTTP-Antwortvergleichsarbeitsgänge (HTTP response comparison operations) als Stapelprozess ausführt. Dementsprechend stellt die vorliegende Erfindung eine leistungsfähige, preiswerte Möglichkeit für eine gründlichere Prüfung einer Web-Inhaltanpassung für mehrere pervasive Datenverarbeitungseinheiten bereit, ohne dass die eigentlichen pervasive Datenverarbeitungseinheiten benötigt werden.

Die vorhergehende Beschreibung veranschaulicht die vorliegende Erfindung und soll nicht als begrenzend gedeutet werden. Obwohl einige beispielhafte Ausführungsformen dieser Erfindung beschrieben wurden, werden Fachleute problemlos verstehen, dass viele Änderungen in den beispielhaften Ausführungsformen möglich sind, ohne wesentlich von der neuen Lehre und den Vorteilen dieser Erfindung abzuweichen. Dementsprechend liegen alle solchen Änderungen im Anwendungsbereich dieser Erfindung, wie sie in den Ansprüchen definiert wird. In den Ansprüchen sind funktionelle Vorrichtungsmerkmale (means-plus-function clause) dafür vorgesehen, die hierin beschriebenen Strukturen bei der Ausführung der genannten Funktion zu umfassen und nicht nur strukturelle Äquivalente, sondern auch funktionelle Äquivalente. Daher muss klar sein, dass die vorhergehende Beschreibung die vorliegende Erfindung

veranschaulicht und nicht als auf die beschriebenen Ausführungsformen begrenzt gedeutet werden soll und dass Änderungen an den beschriebenen Ausführungsformen sowie andere Ausführungsformen im Anwendungsbereich der angehängten Ansprüche enthalten sind. Die Erfindung wird durch die folgenden Ansprüche definiert, wobei Äquivalente der Ansprüche darin enthalten sind.

Patentansprüche

1. Verfahren zum Überprüfen eines für die Anzeige in einer pervasive Datenverarbeitungseinheit angepassten Web-Inhaltes, der auf eine simulierte Anforderung von der pervasive Datenverarbeitungseinheit hin, die von einem Datenverarbeitungssystem erzeugt wurde, das nicht die pervasive Datenverarbeitungseinheit ist, wobei das Verfahren die folgenden Schritte umfasst:

Erstellen einer simulierten HyperText Transfer Protocol-(HTTP-) Anforderung über das Datenverarbeitungssystem, wobei die simulierte HTTP-Anforderung einen Uniform Resource Locator (URL) enthält, der eine Position eines Web-Inhaltes kennzeichnet, und wobei die simulierte HTTP-Anforderung einen HTTP-Vorsatz enthält, der Informationen über die pervasive Datenverarbeitungseinheit enthält;

Ausgeben der simulierten HTTP-Anforderung vom Datenverarbeitungssystem zu einem Web-Server, der in der simulierten HTTP-Anforderung gekennzeichnet wird;

Empfangen einer HTTP-Antwort vom Web-Server im Datenverarbeitungssystem, wobei die HTTP-Antwort einen Web-Inhalt enthält, der von einem Konvertierungs-Proxy, der für die Anpassung eines Web-Inhaltes für die Anzeige in pervasive Datenverarbeitungseinheiten ausgelegt ist, für die Anzeige in der pervasive Datenverarbeitungseinheit angepasst wurde;

Vergleichen der empfangenen HTTP-Antwort mit einer erwarteten HTTP-Antwort.

2. Verfahren nach Anspruch 1, wobei die Informationen über die pervasive Datenverarbeitungseinheit mindestens eine der folgenden Informationen enthalten: eine Kennung eines auf der pervasive Datenverarbeitungseinheit ablaufenden Browsers, eine Kennung eines auf der pervasive Datenverarbeitungseinheit ablaufenden Betriebssystems, eine Kennung von Eigenschaften einer Anzeige für die pervasive Datenverarbeitungseinheit und eine Kennung des Web-Inhaltes, für dessen Anzeige die pervasive Datenverarbeitungseinheit konfiguriert ist.

3. Verfahren nach Anspruch 1, das außerdem den Schritt des Speicherns der auf die simulierte HTTP-Anforderung hin empfangenen HTTP-Antwort umfasst.

4. Verfahren nach Anspruch 1, wobei der Schritt des Vergleichens der empfangenen HTTP-Antwort mit einer erwarteten HTTP-Antwort außerdem den Schritt des Speicherns der empfangenen HTTP-Antwort umfasst, falls die empfangene HTTP-Antwort nicht mit der erwarteten HTTP-Antwort übereinstimmt.

5. Verfahren nach Anspruch 1, wobei der Schritt des Erstellens einer simulierten HTTP-Anforderung über das Datenverarbeitungssystem Folgendes umfasst:

Erstellen einer Vielzahl von simulierten HTTP-Anforderungen von einer entsprechenden Vielzahl pervasive Datenverarbeitungseinheiten über das Datenverarbeitungssystem, wobei jede simulierte HTTP-Anforderung einen Uniform Resource Locator (URL) enthält, der eine Position eines Web-Inhaltes kennzeichnet, und

wobei jede simulierte HTTP-Anforderung einen HTTP-Vorsatz enthält, der Informationen über eine entsprechende pervasive Datenverarbeitungseinheit enthält; und

Speichern der Vielzahl simulierter HTTP-Anforderungen. 5

6. Verfahren nach Anspruch 5, wobei der Schritt des Ausgebens der simulierten HTTP-Anforderung über das Datenverarbeitungssystem zu einem in der simulierten HTTP-Anforderung gekennzeichneten Web-Server das asynchrone Ausgeben der Vielzahl simulierter HTTP-Anforderungen über das Datenverarbeitungssystem an in jeder entsprechenden simulierten HTTP-Anforderung gekennzeichnete Web-Server umfasst. 10

7. Verfahren nach Anspruch 5, wobei der Schritt des Erstellens einer Vielzahl von simulierten HTTP-Anforderungen über das Datenverarbeitungssystem das Abrufen von HTTP-Vorsatzfeldern und URLs aus den jeweiligen Datendateien umfasst. 15

8. Verfahren nach Anspruch 5, wobei der Schritt des Vergleichens der empfangenen HTTP-Antwort mit einer erwarteten HTTP-Antwort das Vergleichen jeder empfangenen HTTP-Antwort mit einer erwarteten HTTP-Antwort für eine pervasive Datenverarbeitungseinheit umfasst, die jeder entsprechenden simulierten HTTP-Anforderung zugeordnet ist. 20

9. Verfahren zum Überprüfen eines für die Anzeige in einer Vielzahl von pervasive Datenverarbeitungseinheiten angepassten Web-Inhaltes, der auf simulierte Anforderungen von den pervasive Datenverarbeitungseinheiten hin, die von einem Datenverarbeitungssystem erzeugt werden, das nicht eine der pervasive Datenverarbeitungseinheiten ist, wobei das Verfahren die folgenden Schritte umfasst: 30

Erstellen einer Vielzahl von simulierten HyperText Transfer Protocol- (HTTP-) Anforderungen über das Datenverarbeitungssystem, wobei jede simulierte HTTP-Anforderung einen Uniform Resource Locator (URL) enthält, der eine Position eines Web-Inhaltes kennzeichnet, und wobei jede simulierte HTTP-Anforderung einen HTTP-Vorsatz enthält, der Informationen über eine entsprechende pervasive Datenverarbeitungseinheit enthält; 35

asynchrones Ausgeben jeder simulierten HTTP-Anforderung vom Datenverarbeitungssystem zu einem Web-Server, der in jeder entsprechenden simulierten HTTP-Anforderung gekennzeichnet wird;

Empfangen einer HTTP-Antwort auf jede entsprechende simulierte HTTP-Anforderung im Datenverarbeitungssystem, wobei jede HTTP-Antwort einen Web-Inhalt enthält, der für die Anzeige in einer pervasive Datenverarbeitungseinheit angepasst wurde, die in einer entsprechenden simulierten HTTP-Anforderung gekennzeichnet wird; und 50

Vergleichen jeder empfangenen HTTP-Antwort mit einer erwarteten HTTP-Antwort für eine pervasive Datenverarbeitungseinheit, die in jeder entsprechenden simulierten HTTP-Anforderung angegeben wird.

10. Verfahren nach Anspruch 9, wobei Informationen über pervasive Datenverarbeitungseinheiten in jeder simulierten HTTP-Anforderung mindestens eine der folgenden Informationen enthalten: eine Kennung eines auf der pervasive Datenverarbeitungseinheit ablaufenden Browsers, eine Kennung eines auf der pervasive Datenverarbeitungseinheit ablaufenden Betriebssystems, eine Kennung von Eigenschaften der Anzeige einer pervasive Datenverarbeitungseinheit und eine 60

Kennung des Web-Inhaltes, für dessen Anzeige die pervasive Datenverarbeitungseinheit konfiguriert ist. 11. Verfahren nach Anspruch 9, das außerdem den Schritt des Speicherns der auf die simulierten HTTP-Anforderungen hin empfangenen HTTP-Antworten umfasst.

12. Verfahren nach Anspruch 9, wobei der Schritt des Vergleichens jeder empfangenen HTTP-Antwort mit einer erwarteten HTTP-Antwort das Speichern der empfangenen HTTP-Antwort umfasst, falls die empfangene HTTP-Antwort nicht mit der erwarteten HTTP-Antwort übereinstimmt.

13. Verfahren nach Anspruch 9, wobei der Schritt des Erstellens einer Vielzahl von simulierten HTTP-Anforderungen das Abrufen von HTTP-Vorsatzfeldern und URLs aus jeweiligen Datendateien umfasst.

14. System, das einen für die Anzeige in einer pervasive Datenverarbeitungseinheit angepassten Web-Inhalt überprüft, der auf eine simulierte Anforderung von der pervasive Datenverarbeitungseinheit hin, die von einem Datenverarbeitungssystem erzeugt wurden, das nicht die pervasive Datenverarbeitungseinheit ist, wobei das System Folgendes umfasst:

Mittel zum Erstellen einer simulierten HyperText Transfer Protocol- (HTTP-) Anforderung, wobei die simulierte HTTP-Anforderung einen Uniform Resource Locator (URL) enthält, der eine Position eines Web-Inhaltes kennzeichnet, und wobei die simulierte HTTP-Anforderung einen HTTP-Vorsatz enthält, der Informationen über die pervasive Datenverarbeitungseinheit enthält;

Mittel zum Ausgeben der simulierten HTTP-Anforderung vom Datenverarbeitungssystem zu einem Web-Server, der in der simulierten HTTP-Anforderung gekennzeichnet wird;

Mittel zum Empfangen einer HTTP-Antwort vom Web-Server im Datenverarbeitungssystem, wobei die HTTP-Antwort einen Web-Inhalt enthält, der von einem Konvertierungs-Proxy, der für die Anpassung eines Web-Inhaltes für die Anzeige in pervasive Datenverarbeitungseinheiten ausgelegt ist, für die Anzeige in der pervasive Datenverarbeitungseinheit angepasst wurde; und

Mittel zum Vergleichen der empfangenen HTTP-Antwort mit einer erwarteten HTTP-Antwort.

15. System nach Anspruch 14, wobei die Informationen über die pervasive Datenverarbeitungseinheit mindestens eine der folgenden Informationen enthalten: eine Kennung eines auf der pervasive Datenverarbeitungseinheit ablaufenden Browsers, eine Kennung eines auf der pervasive Datenverarbeitungseinheit ablaufenden Betriebssystems, eine Kennung von Eigenschaften einer Anzeige für die pervasive Datenverarbeitungseinheit und eine Kennung des Web-Inhaltes, für dessen Anzeige die pervasive Datenverarbeitungseinheit konfiguriert ist.

16. System nach Anspruch 14, wobei das Mittel zum Empfangen einer HTTP-Antwort vom Web-Server ein Mittel zum Speichern der HTTP-Antwort umfasst.

17. System nach Anspruch 14, wobei das Mittel zum Vergleichen der empfangenen HTTP-Antwort mit einer erwarteten HTTP-Antwort ein Mittel zum Speichern der empfangenen HTTP-Antwort umfasst, falls die empfangene HTTP-Antwort nicht mit der erwarteten HTTP-Antwort übereinstimmt.

18. System nach Anspruch 14, wobei das Mittel zum Erstellen einer simulierten HTTP-Anforderung Folgendes umfasst:

Mittel zum Erstellen einer Vielzahl von simulierten HTTP-Anforderungen von einer entsprechenden Vielzahl an pervasive Datenverarbeitungseinheiten, wobei jede simulierte HTTP-Anforderung einen Uniform Resource Locator (URL) enthält, der eine Position eines Web-Inhaltes kennzeichnet, und wobei jede simulierte HTTP-Anforderung einen HTTP-Vorsatz enthält, der Informationen über eine entsprechende pervasive Datenverarbeitungseinheit enthält; und
 5 Mittel zum Speichern der Vielzahl simulierter HTTP-Anforderungen.

19. System nach Anspruch 18, wobei das Mittel zum Ausgeben der simulierten HTTP-Anforderung über das Datenverarbeitungssystem zu einem in der simulierten HTTP-Anforderung gekennzeichneten Web-Server ein
 15 Mittel zum asynchronen Ausgeben der Vielzahl simulierter HTTP-Anforderungen an in jeder entsprechenden simulierten HTTP-Anforderung gekennzeichnete Web-Server umfasst.

20. System nach Anspruch 18, wobei das Mittel zum Erstellen einer Vielzahl von simulierten HTTP-Anforderungen über das Datenverarbeitungssystem ein Mittel zum Abrufen von HTTP-Vorsatzfeldern und URLs aus entsprechenden Datendateien umfasst.

21. System nach Anspruch 18, wobei das Mittel zum Vergleichen der empfangenen HTTP-Antwort mit einer erwarteten HTTP-Antwort ein Mittel zum Vergleichen jeder empfangenen HTTP-Antwort mit einer erwarteten HTTP-Antwort für eine pervasive Datenverarbeitungseinheit umfasst, die jeder entsprechenden simulierten HTTP-Anforderung zugeordnet ist.

22. System, das einen für die Anzeige in einer Vielzahl von pervasive Datenverarbeitungseinheiten angepassten Web-Inhalt überprüft, der auf simulierte Anforderungen von den pervasive Datenverarbeitungseinheiten hin, die von einem Datenverarbeitungssystem erzeugt werden, das nicht eine der pervasive Datenverarbeitungseinheiten ist, wobei das System Folgendes umfasst:

Mittel zum Erstellen einer Vielzahl von simulierten HyperText Transfer Protocol- (HTTP-) Anforderungen, wobei jede simulierte HTTP-Anforderung einen Uniform Resource Locator (URL) enthält, der eine Position eines Web-Inhaltes kennzeichnet, und wobei jede simulierte HTTP-Anforderung einen HTTP-Vorsatz enthält, der Informationen über eine entsprechende pervasive Datenverarbeitungseinheit enthält; Mittel zum asynchronen Ausgeben jeder simulierten HTTP-Anforderung vom Datenverarbeitungssystem zu einem Web-Server, der in jeder entsprechenden simulierten HTTP-Anforderung gekennzeichnet wird; Mittel zum Empfangen einer HTTP-Antwort auf jede entsprechende simulierte HTTP-Anforderung, wobei jede HTTP-Antwort einen Web-Inhalt enthält, der für die Anzeige in einer pervasive Datenverarbeitungseinheit angepasst wurde, die in einer entsprechenden simulierten HTTP-Anforderung gekennzeichnet wird; und

Mittel zum Vergleichen jeder empfangenen HTTP-Antwort mit einer erwarteten HTTP-Antwort für eine pervasive Datenverarbeitungseinheit, die in jeder entsprechenden simulierten HTTP-Anforderung angegeben wird.

23. System nach Anspruch 22, wobei die Informationen über die pervasive Datenverarbeitungseinheit in jeder simulierten HTTP-Anforderung mindestens eine der folgenden Informationen enthalten: eine Kennung eines auf einer pervasive Datenverarbeitungseinheit

ablaufenden Browsers, eine Kennung eines auf einer pervasive Datenverarbeitungseinheit ablaufenden Betriebssystems, eine Kennung von Eigenschaften der Anzeige einer pervasive Datenverarbeitungseinheit und eine Kennung des Web-Inhaltes, für dessen Anzeige eine pervasive Datenverarbeitungseinheit konfiguriert ist.

24. System nach Anspruch 22, wobei das Mittel zum Empfangen einer HTTP-Antwort auf jede entsprechende simulierte HTTP-Anforderung ein Mittel zum Speichern der HTTP-Antworten umfasst.

25. System nach Anspruch 22, wobei das Mittel zum Vergleichen jeder empfangenen HTTP-Antwort mit einer erwarteten HTTP-Antwort ein Mittel zum Speichern der empfangenen HTTP-Antwort umfasst, falls die empfangene HTTP-Antwort nicht mit der erwarteten HTTP-Antwort übereinstimmt.

26. System nach Anspruch 22, wobei das Mittel zum Erstellen einer Vielzahl von simulierten HTTP-Anforderungen ein Mittel zum Abrufen von HTTP-Vorsatzfeldern und URLs aus entsprechenden Datendateien umfasst.

27. Computerprogrammprodukt zum Überprüfen eines für die Anzeige in einer pervasive Datenverarbeitungseinheit angepassten Web-Inhaltes, der auf eine simulierte Anforderung von der pervasive Datenverarbeitungseinheit hin, die von einem Datenverarbeitungssystem erzeugt wird, das nicht die pervasive Datenverarbeitungseinheit ist, wobei das Computerprogrammprodukt ein in einem Computer verwendbares Speichermedium mit einem im Medium integrierten computerlesbaren Programmcode umfasst, wobei das computerlesbare Programmcode Folgendes umfasst:

ein computerlesbares Programmcode zum Erstellen einer simulierten HyperText Transfer Protocol- (HTTP-) Anforderung, wobei die simulierte HTTP-Anforderung einen Uniform Resource Locator (URL) enthält, der eine Position eines Web-Inhaltes kennzeichnet, und wobei die simulierte HTTP-Anforderung einen HTTP-Vorsatz enthält, der Informationen über die pervasive Datenverarbeitungseinheit enthält;

ein computerlesbares Programmcode zum Ausgeben der simulierten HTTP-Anforderung vom Datenverarbeitungssystem zu einem Web-Server, der in der simulierten HTTP-Anforderung gekennzeichnet wird; ein computerlesbares Programmcode zum Empfangen einer HTTP-Antwort vom Web-Server im Datenverarbeitungssystem, wobei die HTTP-Antwort einen Web-Inhalt enthält, der von einem Konvertierungs-Proxy, der für die Anpassung eines Web-Inhaltes für die Anzeige in pervasive Datenverarbeitungseinheiten ausgelegt ist, für die Anzeige in der pervasive Datenverarbeitungseinheit angepasst wurde; und ein computerlesbares Programmcode zum Vergleichen der empfangenen HTTP-Antwort mit einer erwarteten HTTP-Antwort.

28. Computerprogrammprodukt nach Anspruch 27, wobei die Informationen über die pervasive Datenverarbeitungseinheit mindestens eine der folgenden Informationen enthalten: eine Kennung eines auf der pervasive Datenverarbeitungseinheit ablaufenden Browsers, eine Kennung eines auf der pervasive Datenverarbeitungseinheit ablaufenden Betriebssystems, eine Kennung von Eigenschaften einer Anzeige für die pervasive Datenverarbeitungseinheit und eine Kennung des Web-Inhaltes, für dessen Anzeige die pervasive Datenverarbeitungseinheit konfiguriert ist.

29. Computerprogrammprodukt nach Anspruch 27, wobei das computerlesbare Programmcodemittel zum Empfangen einer HTTP-Antwort vom Web-Server ein computerlesbares Programmcodemittel zum Speichern der HTTP-Antwort umfasst. 5

30. Computerprogrammprodukt nach Anspruch 27, wobei das computerlesbare Programmcodemittel zum Vergleichen der empfangenen HTTP-Antwort mit einer erwarteten HTTP-Antwort ein computerlesbares Programmcodemittel zum Speichern der empfangenen 10 HTTP-Antwort enthält, falls die empfangene HTTP-Antwort nicht mit der erwarteten HTTP-Antwort übereinstimmt.

31. Computerprogrammprodukt nach Anspruch 27, wobei das computerlesbare Programmcodemittel zum 15 Erstellen einer simulierten HTTP-Anforderung Folgendes umfasst:

ein computerlesbares Programmcodemittel zum Erstellen einer Vielzahl von simulierten HTTP-Anforderungen von einer entsprechenden Vielzahl an pervasive 20 Datenverarbeitungseinheiten, wobei jede simulierte HTTP-Anforderung einen Uniform Resource Locator (URL) enthält, der eine Position eines Web-Inhaltes kennzeichnet, und wobei jede simulierte HTTP-Anforderung einen HTTP-Vorsatz enthält, der Informationen 25 über eine entsprechende pervasive Datenverarbeitungseinheit enthält; und

ein computerlesbares Programmcodemittel zum Speichern der Vielzahl simulierter HTTP-Anforderungen. 32. Computerprogrammprodukt nach Anspruch 31, 30 wobei das computerlesbare Programmcodemittel zum Ausgeben der simulierten HTTP-Anforderung über das Datenverarbeitungssystem zu einem in der simulierten HTTP-Anforderung gekennzeichneten Web-Server ein computerlesbares Programmcodemittel zum asynchronen Ausgeben der Vielzahl von simulierten HTTP-Anforderungen an in jeder entsprechenden simulierten 35 HTTP-Anforderung gekennzeichnete Web-Server umfasst.

33. Computerprogrammprodukt nach Anspruch 31, 40 wobei das computerlesbare Programmcodemittel zum Erstellen einer Vielzahl von simulierten HTTP-Anforderungen über das Datenverarbeitungssystem ein computerlesbares Programmcodemittel zum Abrufen von HTTP-Vorsatzfeldern und URLs aus entsprechenden 45 Datendateien umfasst.

34. Computerprogrammprodukt nach Anspruch 31, 50 wobei das computerlesbare Programmcodemittel zum Vergleichen der empfangenen HTTP-Antwort mit einer erwarteten HTTP-Antwort ein computerlesbares Programmcodemittel zum Vergleichen jeder empfangenen HTTP-Antwort mit einer erwarteten HTTP-Antwort für eine pervasive Datenverarbeitungseinheit umfasst, die jeder entsprechenden simulierten HTTP-Anforderung zugeordnet ist. 55

35. Computerprogrammprodukt, das einen für die Anzeige in einer Vielzahl von pervasive Datenverarbeitungseinheiten angepassten Web-Inhalt überprüft, der auf simulierte Anforderungen von den pervasive Datenverarbeitungseinheiten hin, die von einem Datenverarbeitungssystem erzeugt werden, das nicht eine der pervasive Datenverarbeitungseinheiten ist, wobei das Computerprogrammprodukt ein in einem Computer verwendbares Speichermedium mit einem im Medium integrierten computerlesbaren Programmcodemittel 60 umfasst, wobei das computerlesbare Programmcodemittel Folgendes umfasst:

ein computerlesbares Programmcodemittel zum Erstellen

len einer Vielzahl von simulierten HyperText Transfer Protocol- (HTTP-) Anforderungen, wobei jede simulierte HTTP-Anforderung einen Uniform Resource Locator (URL) enthält, der eine Position eines Web-Inhaltes kennzeichnet, und wobei jede simulierte HTTP-Anforderung einen HTTP-Vorsatz enthält, der Informationen über eine entsprechende pervasive Datenverarbeitungseinheit enthält;

ein computerlesbares Programmcodemittel zum asynchronen Ausgeben jeder simulierten HTTP-Anforderung vom Datenverarbeitungssystem zu einem Web-Server, der in jeder entsprechenden simulierten HTTP-Anforderung gekennzeichnet wird;

ein computerlesbares Programmcodemittel zum Empfangen einer HTTP-Antwort auf jede entsprechende simulierte HTTP-Anforderung, wobei jede HTTP-Antwort einen Web-Inhalt enthält, der für die Anzeige in einer pervasive Datenverarbeitungseinheit angepasst wurde, die in einer entsprechenden simulierten HTTP-Anforderung gekennzeichnet wird; und

ein computerlesbares Programmcodemittel zum Vergleichen jeder empfangenen HTTP-Antwort mit einer erwarteten HTTP-Antwort für eine pervasive Datenverarbeitungseinheit, die in jeder entsprechenden simulierten HTTP-Anforderung angegeben wird.

36. Computerprogrammprodukt nach Anspruch 35, wobei die Informationen über die pervasive Datenverarbeitungseinheit in jeder simulierten HTTP-Anforderung mindestens eine der folgenden Informationen enthalten: eine Kennung eines auf einer pervasive Datenverarbeitungseinheit ablaufenden Browsers, eine Kennung eines auf einer pervasive Datenverarbeitungseinheit ablaufenden Betriebssystems, eine Kennung von Eigenschaften der Anzeige einer pervasive Datenverarbeitungseinheit und eine Kennung des Web-Inhaltes, für dessen Anzeige eine pervasive Datenverarbeitungseinheit konfiguriert ist.

37. Computerprogrammprodukt nach Anspruch 35, wobei das computerlesbare Programmcodemittel zum Empfangen einer HTTP-Antwort auf jede entsprechende simulierte HTTP-Anforderung ein computerlesbares Programmcodemittel zum Speichern der HTTP-Antworten umfasst.

38. Computerprogrammprodukt nach Anspruch 35, wobei das computerlesbare Programmcodemittel zum Vergleichen jeder empfangenen HTTP-Antwort mit einer entsprechenden erwarteten HTTP-Antwort ein computerlesbares Programmcodemittel zum Speichern der empfangenen HTTP-Antwort enthält, falls die empfangene HTTP-Antwort nicht mit der erwarteten HTTP-Antwort übereinstimmt.

39. Computerprogrammprodukt nach Anspruch 35, wobei das computerlesbare Programmcodemittel zum Erstellen einer Vielzahl von simulierten HTTP-Anforderungen ein computerlesbares Programmcodemittel zum Abrufen von HTTP-Vorsatzfeldern und URLs aus entsprechenden Datendateien umfasst.

Hierzu 5 Seite(n) Zeichnungen

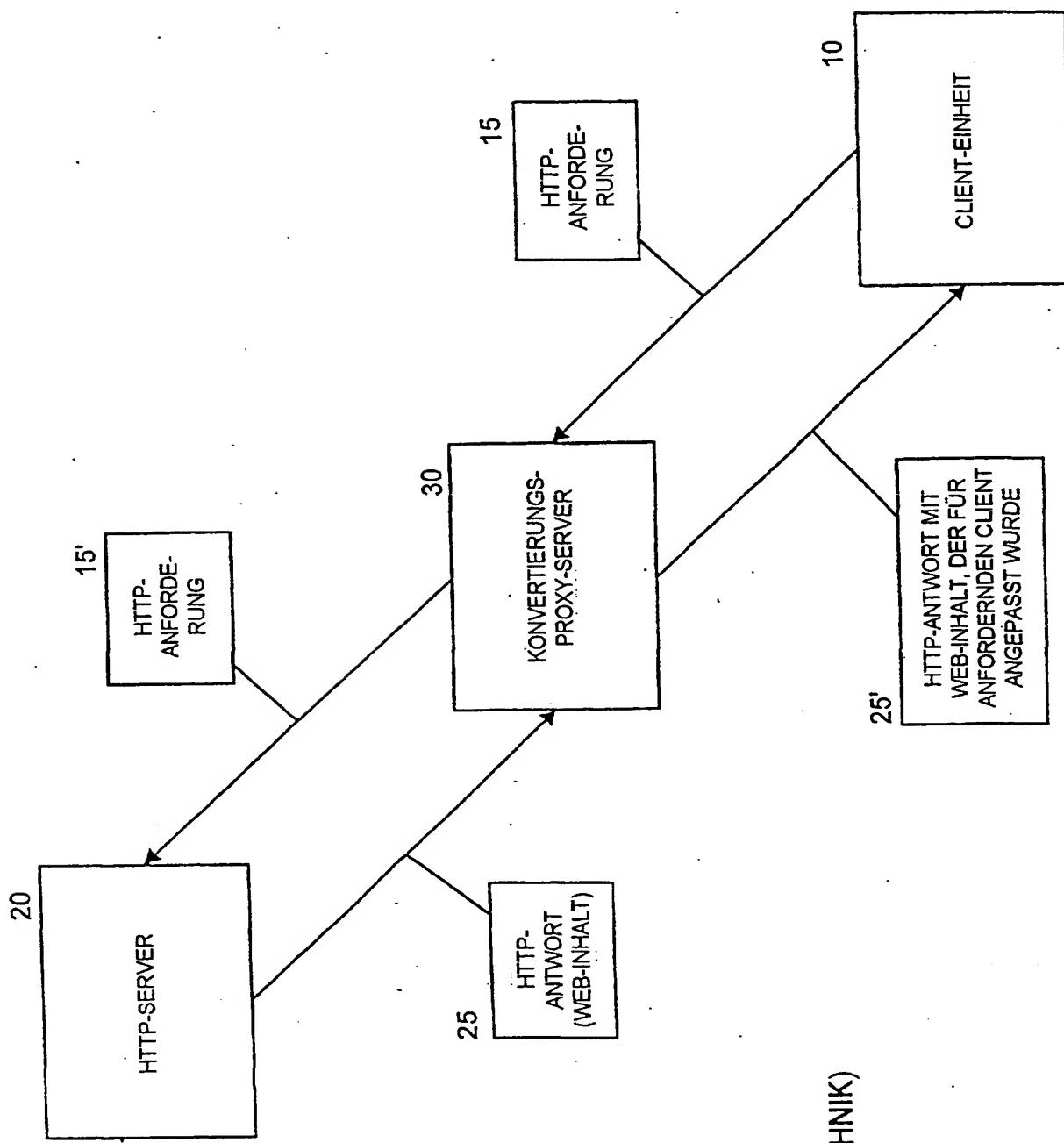


Fig. 1
(STAND DER TECHNIK)

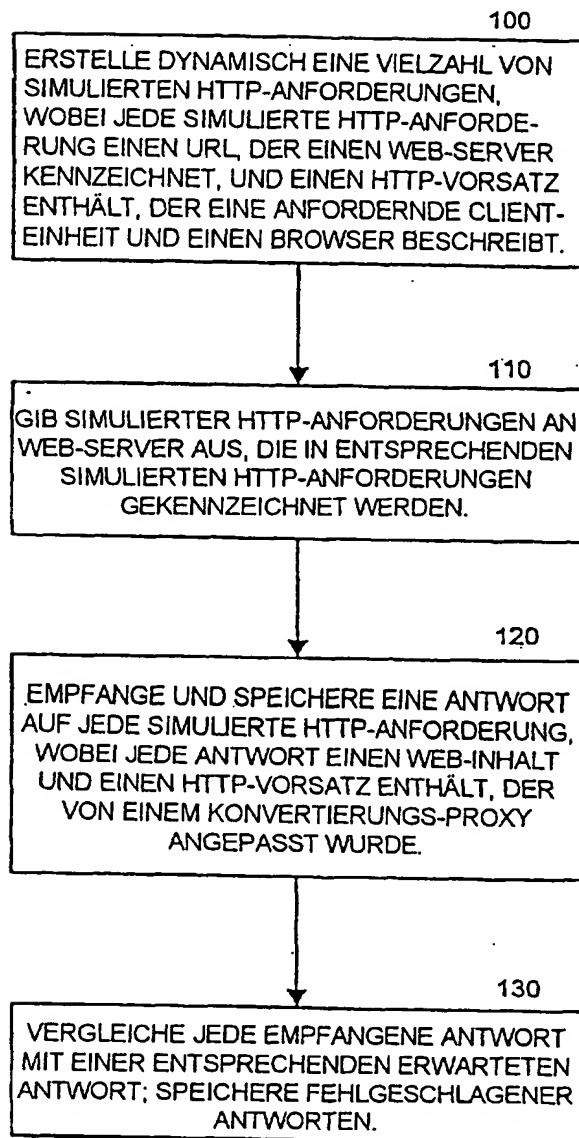
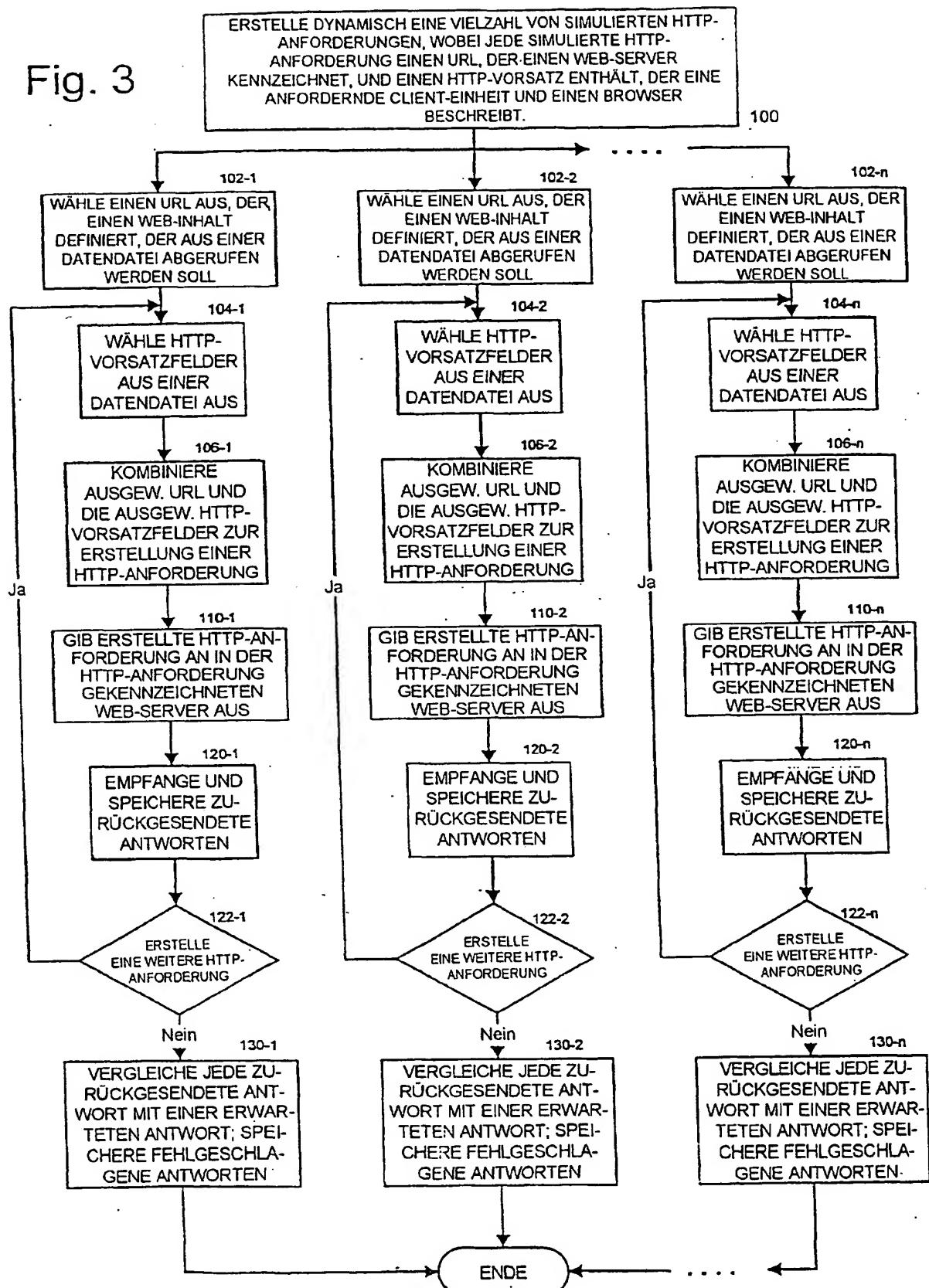


Fig. 2

Fig. 3



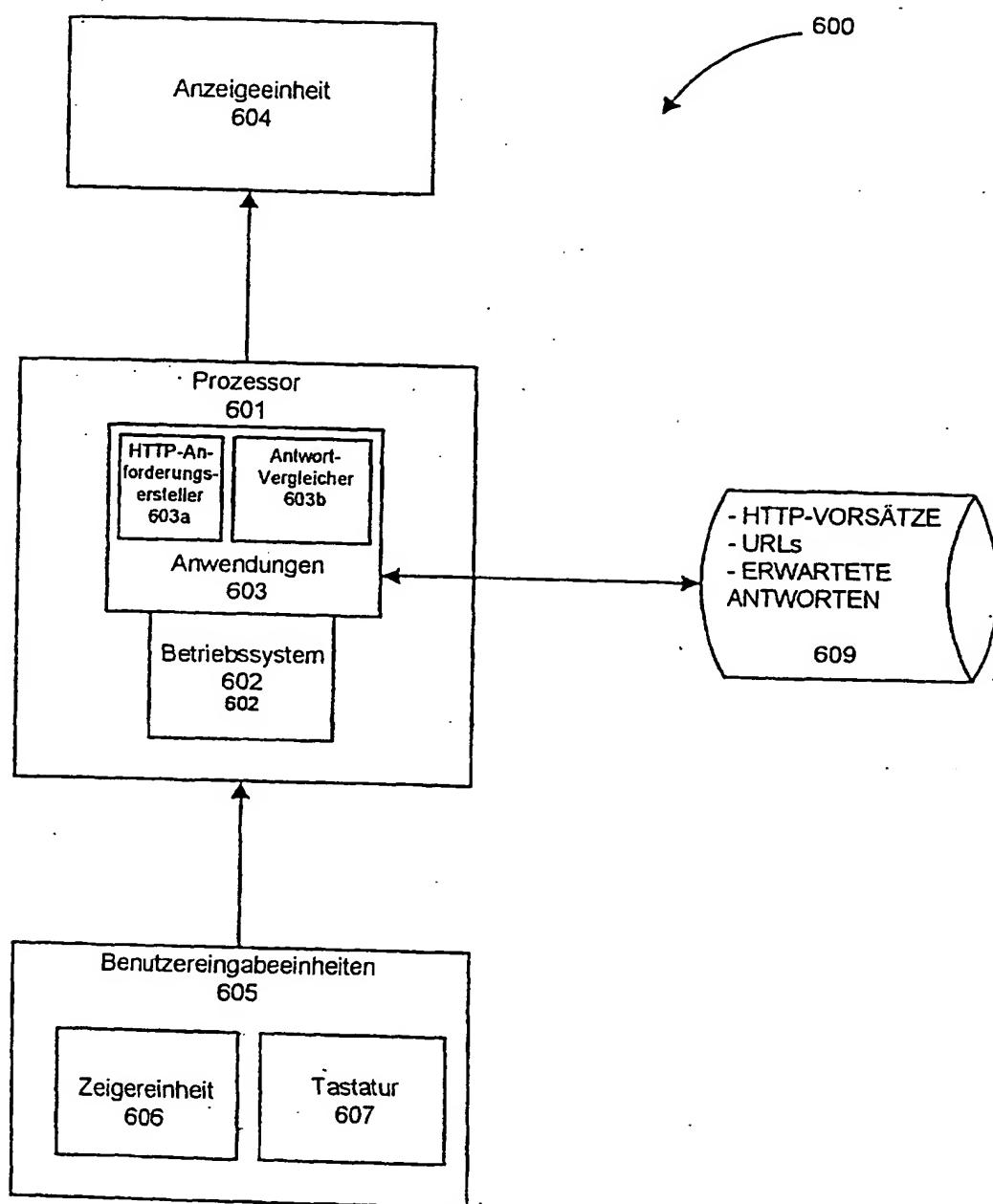


Fig. 4

Fig. 5

